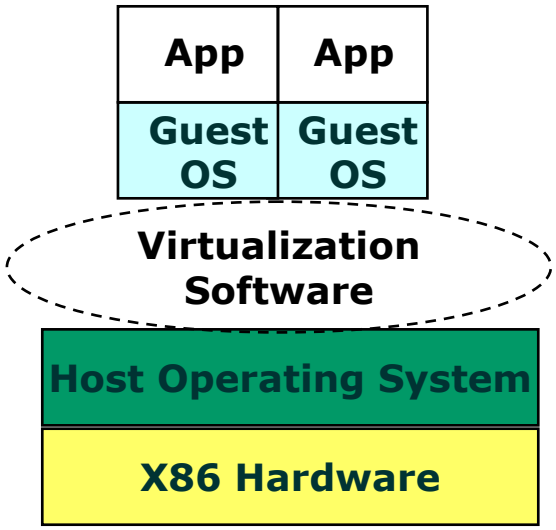
AMD "*Pacifica*" Technology: Enhancements for Virtualization

Fall Processor Forum

Virtual Machine Approaches

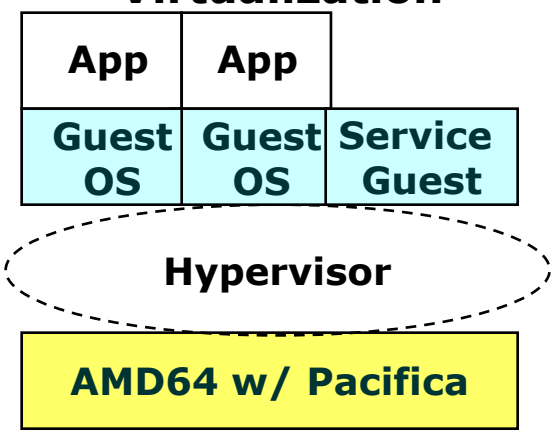
Carve a Platform into Many Virtual Machines

Hosted Virtualization



- Virtualization software manages resources between Host and Guest OS's

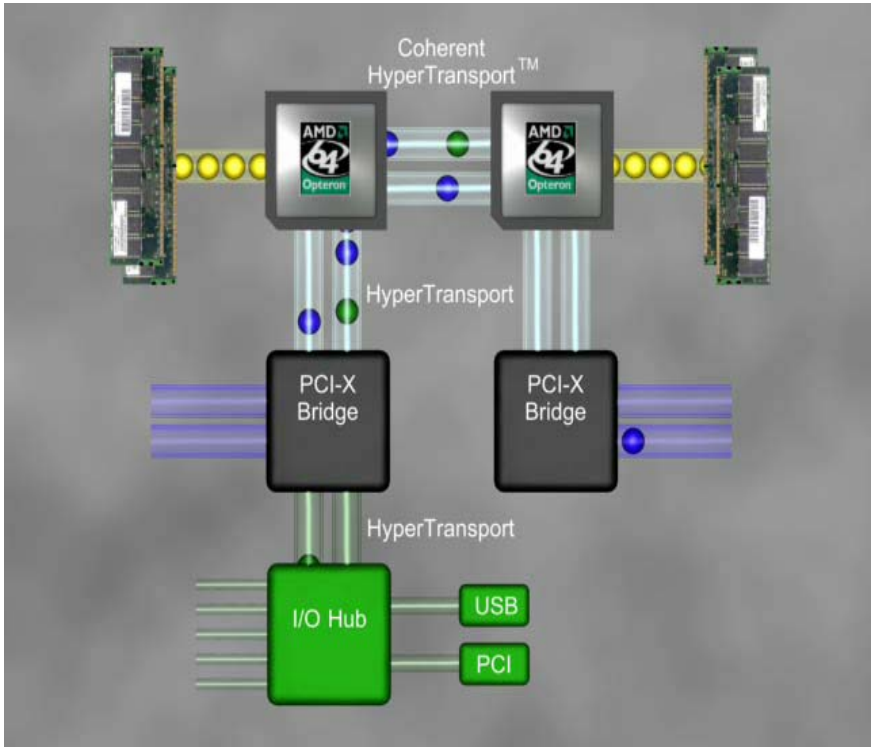
Hypervisor-based Virtualization



- Virtualization Software (Hypervisor) is the host environment.
- Enables better software performance by eliminating some of the associated overhead

System Architecture Makes a Difference

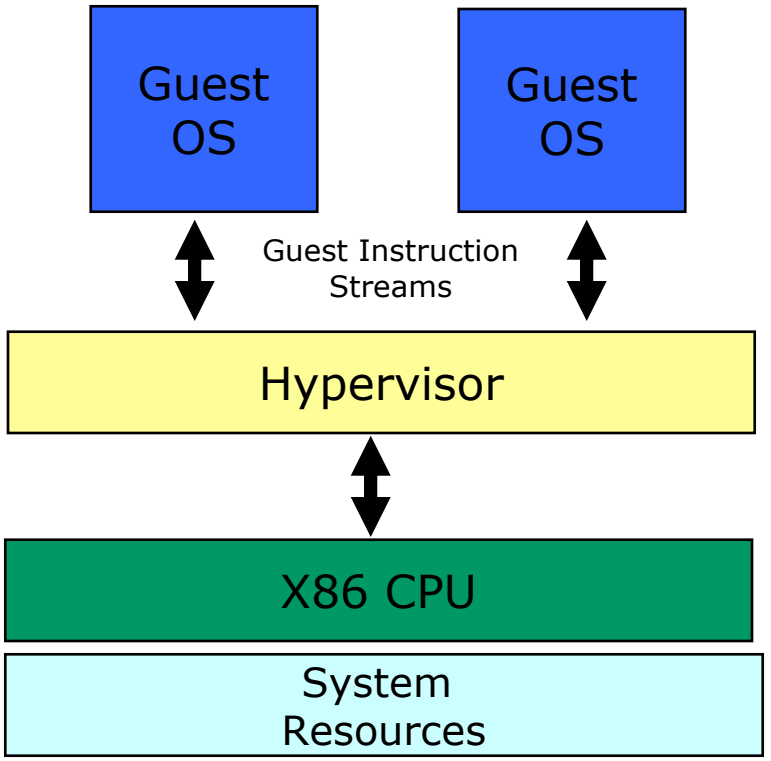
- Multicore, directly-connected design helps reduce the bottlenecks, enabling efficient partitioning



Driving virtualization into the processor ***with Pacifica***!

AMD

- Native virtualization of x86 architecture requires "unnatural acts" to achieve – leading to increased performance overhead, lower security, and increased complexity



Hypervisor monitors instructions, performs dynamic code-rewriting –

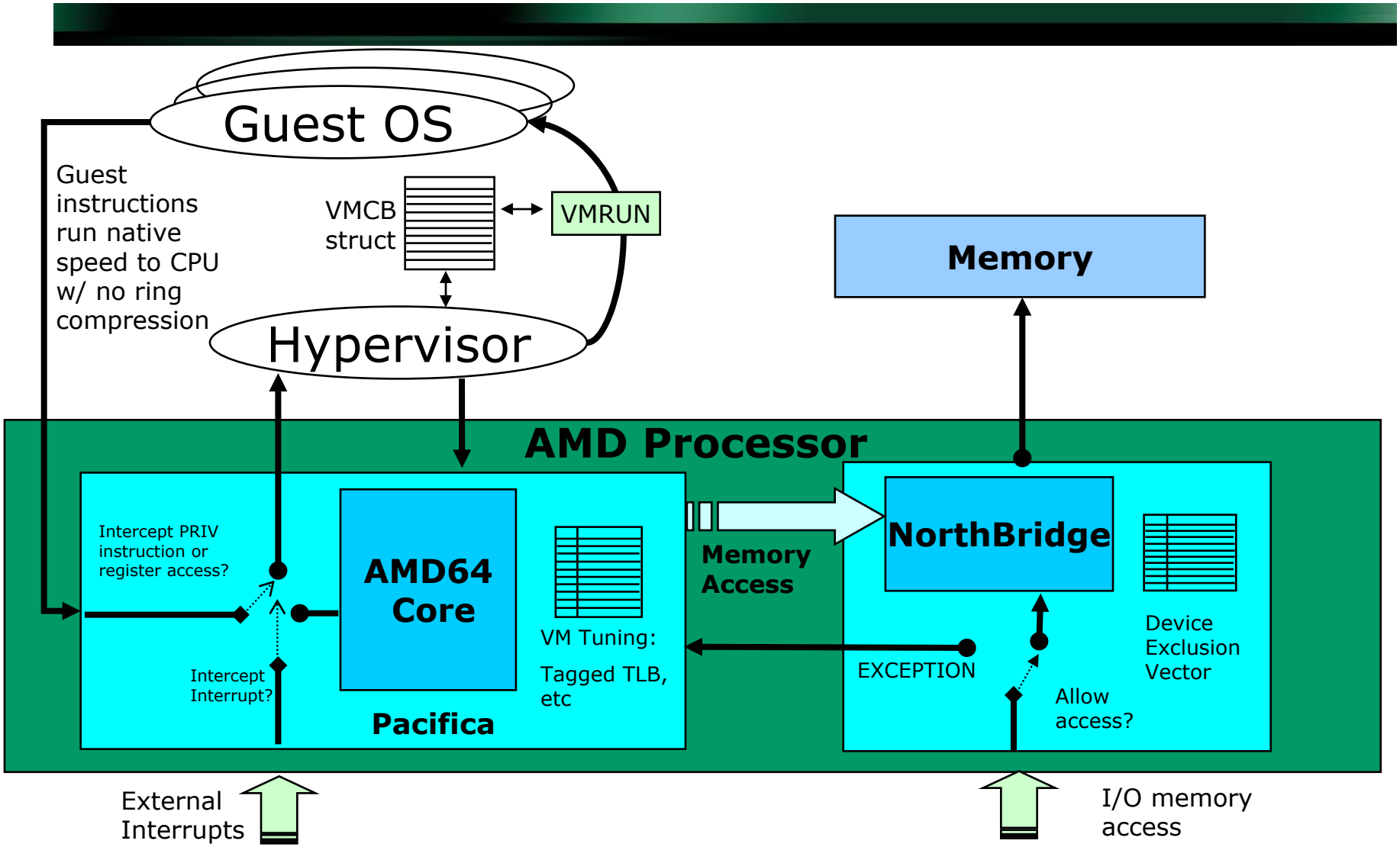
Inefficient!!

Pacifica Features to Accelerate & Secure Virtualization

AMD

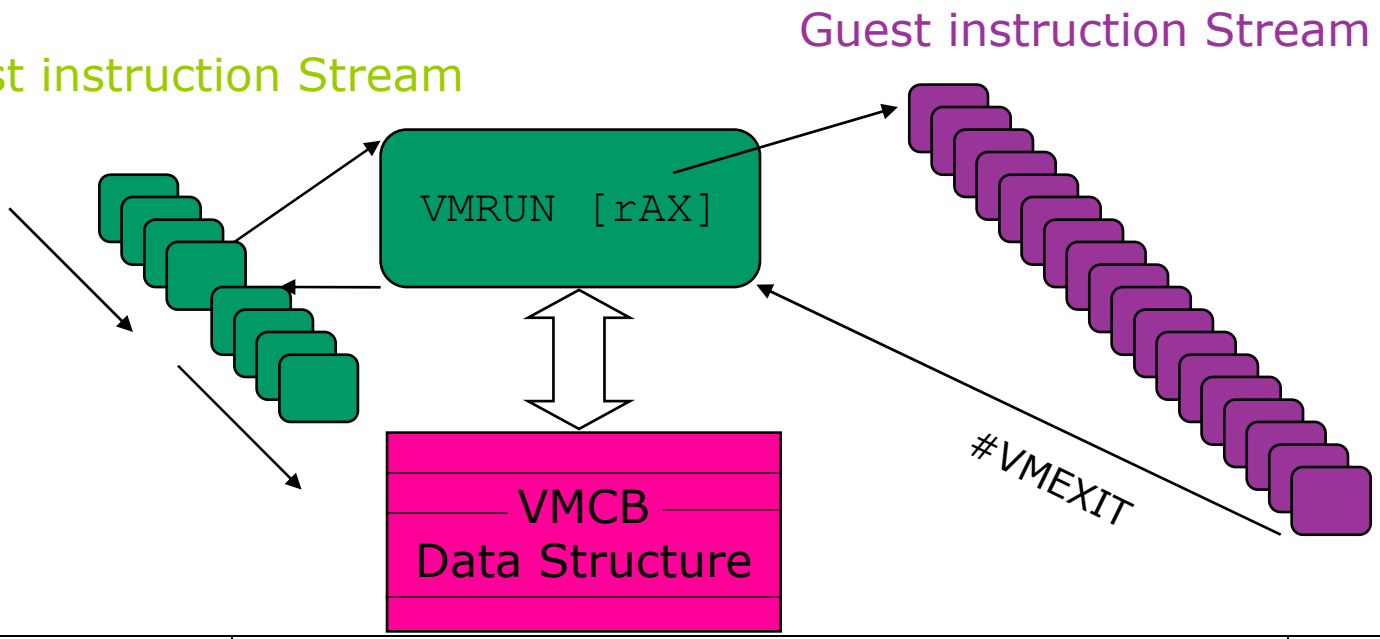
- New Processor Mode: ***Guest Mode***
- New Instruction: ***VMRUN***
- New Data Structure: ***Virtual Machine Control Block (VMCB)***
- Control of Guest execution: ***Selective Interception***
- Enhanced memory management for efficient virtualization
  - ***Nested Page Table Support***
  - ***Tagged TLB***
  - ***Paged Real Mode***
- ***Interrupt architecture enhancements***
- I/O Access Protection through ***Device Exclusion Vectors (DEV)***
- Support for ***SKINIT*** ("secure kernel" init)

Pacifica Silicon Enhanced Virtualization



Pacifica Architecture: VMRUN

- Virtualization based on Virtual Machine Run (VMRUN) instruction
- VMRUN executed by host causes the guest to run
- World-switch: host → guest → host
- Guest runs until it exits back to the host (#VMEXIT)
- Host resumes at the instruction following VMRUN



Pacifica Architecture: Intercepts

- Rich set of intercepts determine what actions cause the guest to exit to hypervisor
  - The VMCB for a guest specifies intercepts
  - Intercepts can vary from guest to guest
  - Two kinds of intercepts
    - Instruction intercepts
    - Exception / Interrupt intercepts
- Guest runs until:
  - It performs an intercept causes an exit to the hypervisor
  - #VMEXIT saves information about the intercepted event into VMCB

Pacifica Architecture: VMCB

- All CPU state for guest is located in the Virtual Memory Control Block (VMCB)data structure
- VMRUN: Entry
  - Host state is saved to memory
  - Guest state loaded from VMCB
  - Guest runs
- VMRUN: Exit
  - Guest state is saved back to VMCB
  - Host state loaded from memory
- Host state saved using Model Specific Register (MSR): `vm_hsave_pa`

Pacifica Architecture: Virtualizing Address Translation

AMD

- Two methods to handle paging /virtual address translation
  - Shadow Page Tables (SPT): hypervisor creates a Shadow Page table by reading guests page tables
  - Nested Paging (NP): the processor walks both host and guest page tables

- Address Space Identifier (ASIDs) improves TLB performance
  - Allows Host and Guest translations to co-exist in TLB
  - Hypervisor assigns each Guest an ASID
    - Specified in VMCB, swapped on world switch
  - ASID is part of the TLB match
  - Eliminates need to flush TLB on world switch

Shadow Page Tables (SPT)

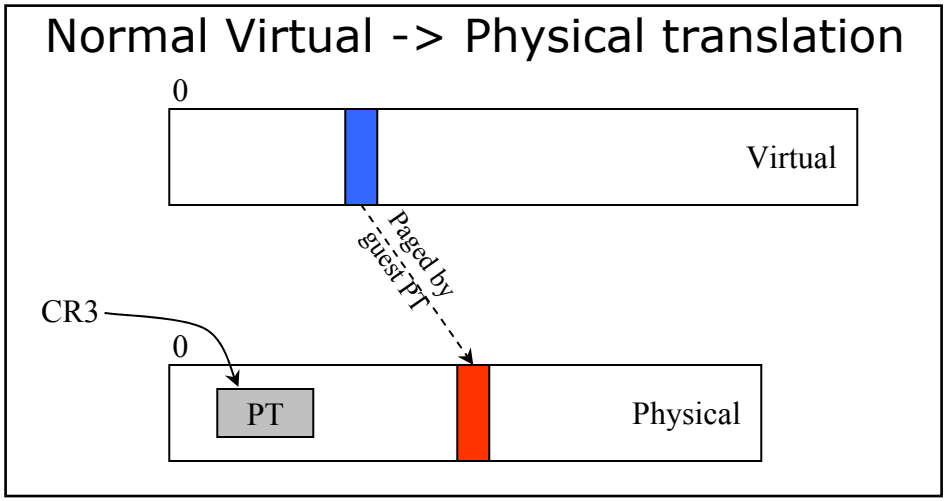
AMD

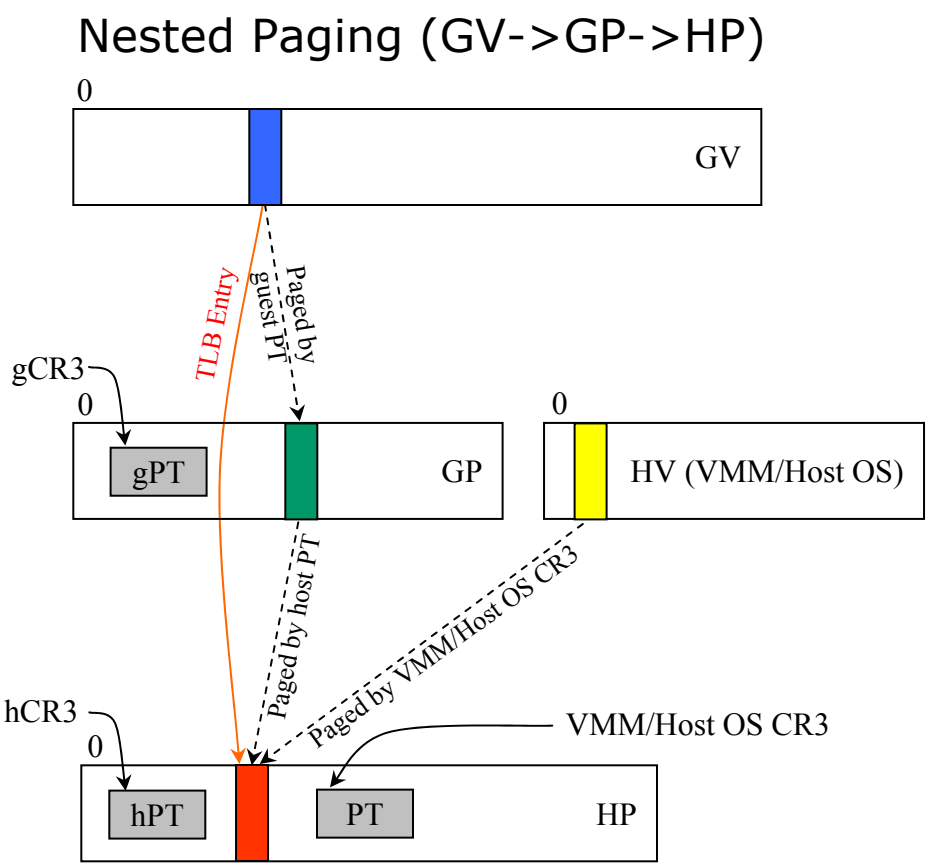
- Hypervisor constructs Shadow Page tables
  - Host intercepts guest page table base register (CR3) Reads/Writes
  - Walks the guest page tables which maps Guest Virtual-> Guest Physical
  - Constructs SPT which maps Guest Virtual -> Host Physical
- Guest runs with CR3 pointing to shadow page tables
  - Guest never sees the "real" (SPT) page tables
  - Or the the "real" contents of CR3
- Hypervisor monitors guest edits to guest page tables
  - Guest page tables are marked "read only"
  - Host reflects guest edits into SPT
  - Also monitors CR3 changes

Nested Paging

- Guest VA translation use both Guest and Host page tables
- Host's paging state underlies guest paging state; CR0, CR2, CR3, paging control bits in EFER are all duplicated.
  - Guest page tables are walked, followed by host page table walk.
  - All accesses by a guest, including reads of guest page table entries, are considered user level in the host.
- Nested paging requires only that the hypervisor have page table mappings for the guest in the hypervisor's own virtual address space; the guest does its own paging.
- NP reduces #VMEXIT frequency and hypervisor complexity.

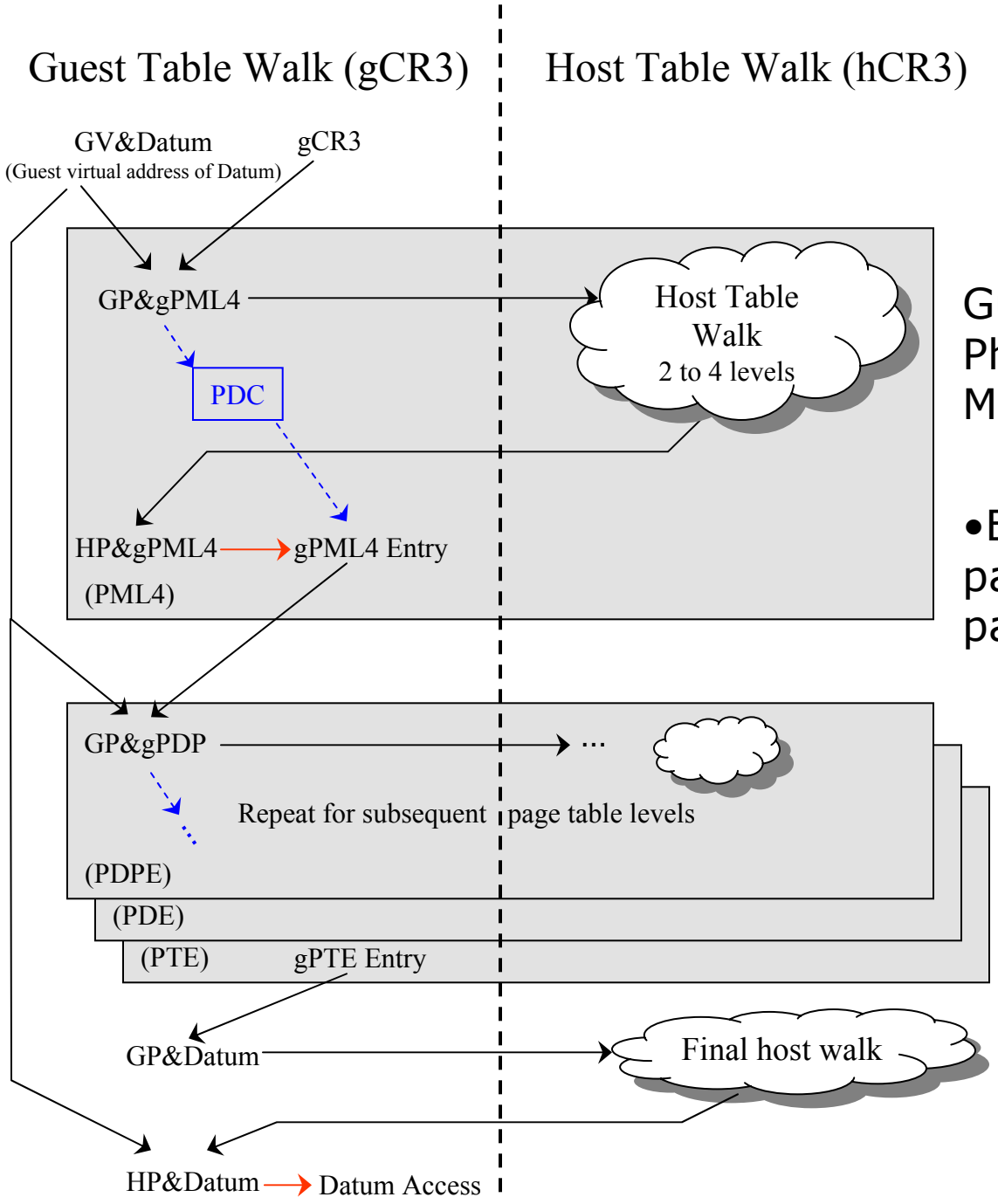
Normal vs. nested paging





GV Guest Virtual
GP Guest Physical
HV Host Virtual
HP Host Physical

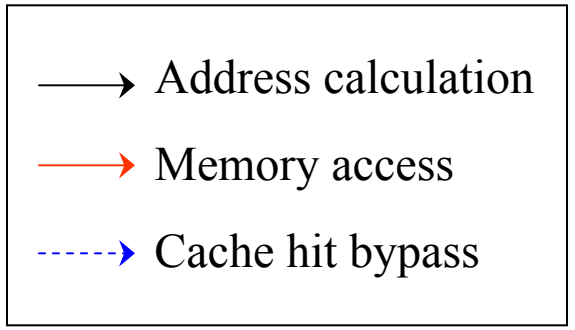
TLB stores final GV->HP translation and optionally stores intermediate translations



AMD

Guest Virtual -> Host Physical in Nested Paging Mode

- Each level of the guest page table requires a host page table walk



Nested Paging Page Faults

AMD

- Faults in the guest translation represent "normal" page faults
  - Delivered to guest as a regular page fault (#PF)
  - No intervention of hypervisor required

- Faults in the host translation are "Nested Paging Faults"
  - causes #VMEXIT to Hypervisor
    - VMCB exit code = VMEXIT_NPF
  - Example: Guest VA 0x2000 ➔ GP 0x50000 ➔ HP 0x150000
    - If hPT entry for 0x50000 marked not present in host: NPF

Pacifica Architecture: Paged Real Mode

- Memory protections rely on paging, guests *must* run with paging enabled
- BIOS and SMM code are designed to start in real mode
- Pacifica Solution: Paged Real Mode
  - Paging enabled in Real Mode
    - Hypervisor specifies CR0.PG=1, CR0.PE=0 in VMCB
  - Real-mode address translation (segment+offset) = Linear address ➔ translation via SPT ➔ physical address
  - Correct composition of SPT's is host responsibility
    - Guest is assuming linear, 0-based mapping

Pacifica Architecture: CPU Access Protection

- Memory Protection via Hypervisor control of page tables
  - No guest access is possible unless a mapping is present in the SPT or nested page tables
- Access to I/O ports controlled by an IO Permission Map (IOPM)
  - VMCB contains a pointer to IOPM that controls guest access rights to IO Ports
  - Granularity is to 1-byte port
- Access to Model Specific Registers (MSR)
  - VMCB contains a pointer to an MSR permission map that control guest access to MSRs
  - Separate read and write intercept bits

Pacifica Architecture: Interrupts

- Processor response to HW interrupts is setup in the VMCB
- Two Options:
  - Hardware interrupts while guest is running are intercepted causing exit to host
    - Host manages physical APIC
    - Host determines interrupt routing and distribution
    - Host injects virtual interrupts into guests as needed
    - Hardware support for virtual interrupts: v_irq, v_vector, v_prio , v_tpr, PHYS_IF
  - Interrupts serviced directly in the guest
    - Guest manages physical APIC
    - Host can still inject virtual interrupts

Pacifica: DMA Protection

- Device Exclusion Vector (DEV) guards physical pages from device access
  - 1 bit per physical 4K page
  - DEV is cached in Northbridge
  - *Contiguous* table in physical memory (128K for 4GB)
- Protection Domains
  - Mapping from bus/device ID to protection domains
  - Multiple protection domains
  - One DEV per protection domain

Future Developments

AMD

- IOMMU
  - translate device addresses to system physical memory addresses
- Faster world switch
  - Faster access to non-renamed architectural state
  - Locally cached guest state
- Hardware interrupt routing
  - Allow guests to directly handle specified interrupt sources
  - S/W interrupts also

Trademark Attribution